

## Persistence Data

### 1 Keeping data within an Activity

For a quick save of status: use the Bundle class

Bundle: is a class for keeping simple key-value pairs.

Example... save & get back:

```
protected void onSaveInstanceState(Bundle outState) {
    outState.putInt("counter", counter);
    Log.i("statesave", "on Save Instance State");
    super.onSaveInstanceState(outState);
}
```

@Override

```
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    counter = savedInstanceState.getInt("counter");
    Log.i("statesave", "onRestoreInstanceState");
    super.onRestoreInstanceState(savedInstanceState);
}
```

These are built-in methods in Activity. You don't have to do anything new here... just override the existing methods in your Activity class.

They make sure your internal data is restored (for example when you change the configuration (perspective of the App)).

### 2 Keeping data within a session

Android provides the `SharedPreferences` object to help you save simple application data.

You can use this in an Activity or in the Application.

It saves a key-value pair and is accessible throughout the application.

Example in an Activity:

Here we write a password into Preferences:

```
/**
 * Write password into preferences.
 * @param result
 */
private void setPreferences(String result){

    prefs = getSharedPreferences("password", MODE_PRIVATE);
    Editor editor = prefs.edit();
    editor.putString("password", result);
    editor.commit();
}
```

With Editor we can change and write the preference.

Here we read the preferences:

```
private boolean checkPreferences(){
    //check if password has already been set:
    prefs = getSharedPreferences("password", MODE_PRIVATE);
    String pw = prefs.getString("password", null);
    if (pw != null && pw.equals("test")){
        Toast.makeText(getApplicationContext(), "Found correct password !",
Toast.LENGTH_LONG).show();
        password = pw;
        return true;
    }
    return false;
}
```

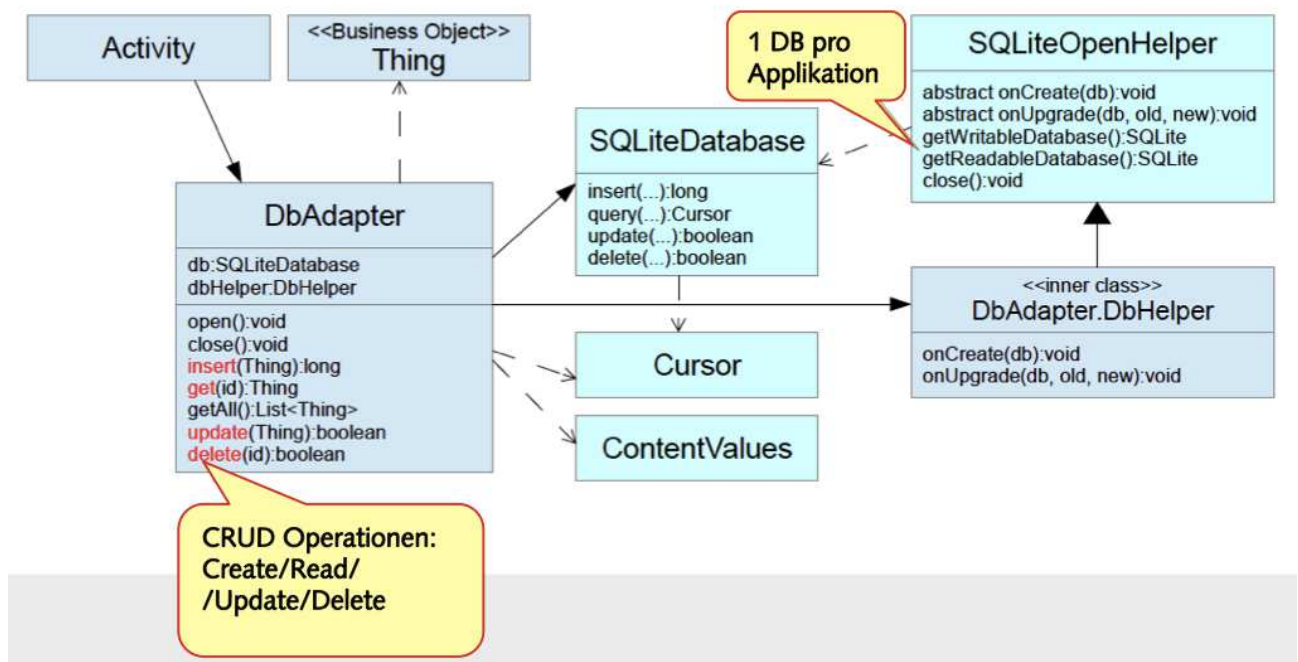
So with the method `getSharedPreferences` you actually create and read the preferences.

Further possibilities: saving to an internal storage in a file (= private). Or to an external storage (SD card) in a file....(= public)

### 3 Using the built-in database

Android has a ready-installed mini database called SQLite.

The database is stored in `data/data/<package-name>/databases` folder.



Best practice:

`DBAdapter` class

Create a `DBAdapter` helper class. This class deals with storing, retrieving, altering the data. It uses the `SQLiteDatabase` class to query, insert, update data....

`DatabaseHelper` class

This class extends the `SQLiteOpenHelper` class. It overrides `onCreate` and `onUpdate`. It can be an inner class in `DBAdapter` or a separate class...

## 4 Using Cloud to store data

Check this page for an overview:

<https://developer.android.com/training/building-connectivity.html>

For data storage, use a Network Connection

For data backups, use the Backup facilities....

According to Android Developer Page there are two ways:

- 1) Configuring Auto Backup
- 2) Using the Backup API (Android 5.1 and lower)

Check out <https://developer.android.com/training/backup/index.html>